

**Beyond the NAS Parallel Benchmarks:
Measuring Dynamic Program Performance
and Grid Computing Applications**

Rob F. Van der Wijngaart (Computer Sciences Corp.)

Rupak Biswas, Michael Frumkin (NAS Division)

Huiyu Feng (George Washington University)

NASA Ames Research Center
Moffett Field, CA

Contents

1. A brief history of NPB
2. What is (not) being measured by NPB?
3. Irregular dynamic applications (UA Benchmark)
4. Wide area distributed computing (NAS Grid Benchmarks – NGB)

A Brief History of NPB

Goal: Measure performance of modern (parallel) architectures running scientific apps

Contents: 5 kernels, 3 pseudo apps (implicit CFD)

Approach: NPB1: Paper-and-pencil specs
NPB2: Source code implementations (F77/C/MPI)

PBN: Source code implementations (HPF/OpenMP/Java)

A Brief History of NPB (cont'd)

- Kernels:**
- EP Random-number generator
 - IS Integer sort
 - CG Conjugate gradient
 - MG Multigrid method for Poisson eqn
 - FT Spectral method (FFT) for Laplace eqn

- Pseudo apps:**
- BT ADI; Block-Tridiagonal systems
 - SP ADI; Scalar Pentadiagonal systems
 - LU Lower-Upper symmetric Gauss-Seidel

What is tested in NPB2?

Name	Math functions	Network bandwidth	Network latency	Memory bandwidth	Instruction cache
EP	✓				
IS		✓		✓	
CG		✓		✓	
MG				✓	
FT	✓	✓			
BT		✓			✓
SP				✓	
LU			✓		✓

What is not tested in NPB2?

- Dynamically changing memory accesses
 - Irregular memory accesses
 - False sharing / cache coherence costs
- } UA
- System software / Grid computing middleware
 - Fault tolerance
 - Wide area (public) network bandwidth/latency
- } NGB

Unstructured Adaptive Mesh Refinement (UA)

- Structured, static grids → fixed-stride memory access
 - Message passing → private address spaces
- Choice of parallelization paradigm not (very) important

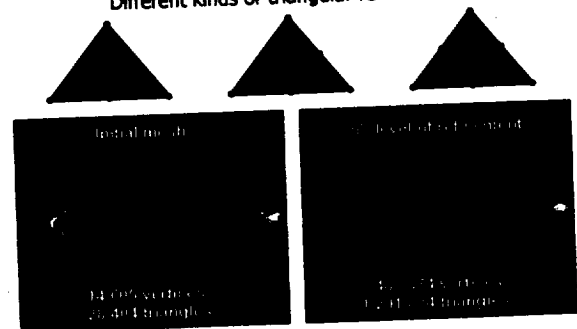
Unstructured, adaptive mesh refinement (Biswas, Olicker)

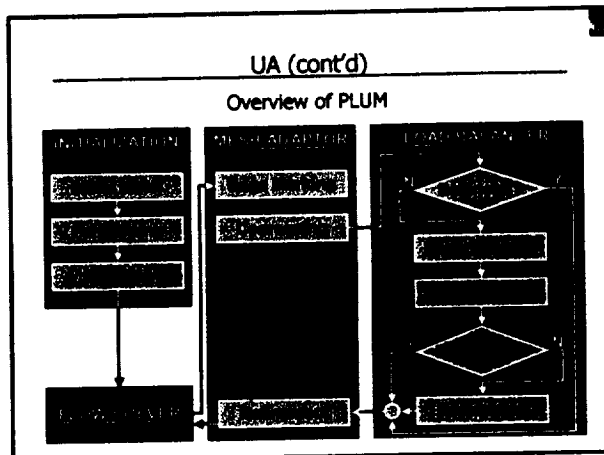
- No solver - only refine, (re)partition, and remap
- Implementations:

- MPI - SGI Origin, Cray T3E
- OpenMP - SGI Origin
- Multithreading - Tera (Cray) MTA

UA (cont'd)

Different kinds of triangular refinement





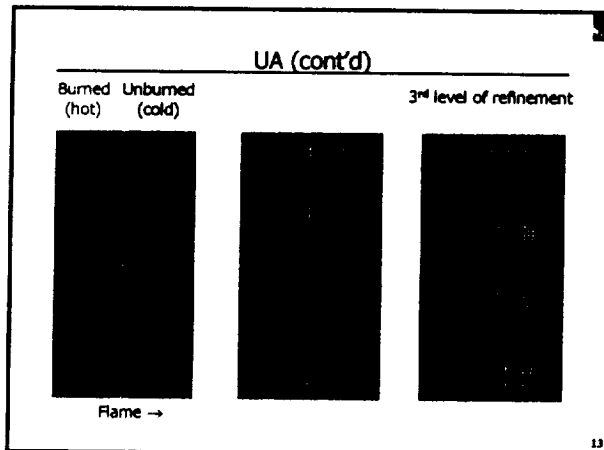
UA (cont'd)

Overall assessment

Paradigm	Code increase	Memory increase	Scalability	Portability
MPI	100%	70%	Medium	High
OpenMP	10%	5%	None	Medium
Multi-threading	2%	7%	High	Low

- UA (cont'd)
- Complete, practically useful benchmark:
- Includes solver
 - Is 3-dimensional
 - Does coarsening in addition to refinement
 - Spends little time on (re)partitioning/(re)mapping
 - Is well load balanced
 - Requires no data files
 - Is compact: NPB2 apps ~ 5,000 lines (SP,BT,LU)

- UA (cont'd)
- Flame propagation problem (Feng & Mavriplis):
- Scalar transport eqn: $T_t + V \cdot \nabla T = \epsilon \Delta T - \epsilon^{-1} F(T)$
 - Velocity field given: linear problem
 - Rectangular domain
 - Rectangular nonconforming elements
 - Spectral elements of relatively high order (5th)
 - Mixed explicit/implicit time integration
 - Spatial refinement/coarsening
 - Interface ops cheap compared to overall scheme



NAS Grid Benchmarks (NGB)

Goal: Measure performance of modern distributed systems, emphasizing Grid computing. Also gauges functionality.

Contents: 4 compound tasks: 3 pseudo apps (implicit CFD), 2 kernels, all from NPB2.

Approach: NGB1: Paper-and-pencil specs.
NGB2: Source code implementations (NPB2/PBN + Globus/Legion/Corba/Condor/Java ...)

14

NGB cont'd

Must define boundaries because:

- Grid concept/environment/infrastructure not well defined
- Grid has very many software/hardware components
- Grid has complex functional hierarchy
- Grid is a time/organization/application dependent target

Cannot freeze benchmark implementation because:

- No clear Grid environment winner
- No clear picture of representative Grid application

15

NGB cont'd

- Provide synthetic Grid app. for scientific computing:
 - data flow graph coupling NPB codes
- Specify:
 - abstract services necessary
 - problem sizes (classes): S, W, A, B, C
- Do not specify:
 - fault recovery reqs
 - security reqs
- Measure and report turnaround time

16